BEST PRACTICE
Structural System Testing Technique Categories

Structural system testing is designed to verify that the developed system and programs work. The objective is to ensure that the product designed is structurally sound and will function correctly. It attempts to determine that the technology has been used properly and that when all the component parts are assembled they function as a cohesive unit. The structural system testing techniques provide the facility for determining that the implemented configuration and its interrelationship of parts functions so that they can perform the intended tasks. The techniques are not designed to ensure that the application system is functionally correct, but rather, that it is structurally sound.
The structural system testing techniques are briefly described in Table 7.

Table 7. Structural Testing Techniques

Technique	Description	Example
Stress	Determine system performs with expected volumes.	- Sufficient disk space allocated - Communication lines adequate
Execution	System achieves desired level of proficiency.	- Transaction turnaround time adequate - Software/hardware use optimized
Recovery	System can be returned to an operational status after a failure.	- Induce failure - Evaluate adequacy of backup data
Operations	System can be executed in a normal operational status.	- Determine systems can run using document - JCL adequate
Compliance (to Process)	System is developed in accordance with standards and procedures.	- Standards followed - Documentation complete
Security	System is protected in accordance with importance to organization.	- Access denied - Procedures in place

Stress Testing Techniques

Stress testing is designed to determine if the system can function when subject to large volumes – larger than would be normally expected. The areas that are stressed include input transactions, internal tables, disk space, output, communications, computer capacity, and interaction with people.

Objectives

The objective of stress testing is to simulate a production environment for the purpose of determining that:

- Normal or above-normal volumes of transactions can be processed through the transaction within the expected time frame.
- The application system is structurally able to process large volumes of data.
- System capacity, including communication lines, has sufficient resources available to meet expected turnaround times.
- People can perform their assigned tasks and maintain the desired turnaround time.

How to Use Stress Testing

Stress testing should simulate as closely as possible the production environment. Online systems should be stress tested by having people enter transactions at a normal or above normal pace. Batch systems can be stress tested with large input batches. Error conditions should be included in tested transactions. Transactions for use in stress testing can be obtained from one of the following three sources:

- Test data generators.
- Test transactions created by the test group.
- Transactions previously processed in the production environment.

Operators should use standard documentation, and the people entering transactions or working with the system should be the clerical personnel that will work with the system after it goes into production. Online systems should be tested for an extended period of time, and batch systems tested using more than one batch of transactions.

Stress Test Example

Stress tests can be designed to test all or parts of an application system. For example, stress testing might:

- Enter transactions to determine that sufficient disk space has been allocated to the application.
- Ensure that the communication capacity is sufficient to handle the volume of work by attempting to overload the network with transactions.
- Test system overflow conditions by entering more transactions than can be accommodated by tables, queues, and internal storage facilities, etc.

When to Use Stress Testing

Stress testing should be used when there is uncertainty regarding the amount of work the application system can handle without failing. Stress testing attempts to

break the system by overloading it with a large volume of transactions. Stress testing is most common with online applications because it is difficult to simulate heavy volume transactions using the other testing techniques. The disadvantage of stress testing is the amount of time it takes to prepare for the test, plus the amount of resources consumed during the actual execution of the test. These costs need to be weighed against the risk of not identifying volume-related failures until the application is placed into an operational mode.

Execution Testing Technique

Execution testing determines whether the system achieves the desired level of proficiency in a production status. Execution testing can verify response times, turnaround times, as well as design performance. The execution of a system can be tested in whole or in part, using the actual system or a simulated model of a system.

Objectives

Execution testing is used to determine whether the system can meet the specific performance criteria. The objectives of execution testing include:

- Determine the performance of the system structure.
- Verify the optimum use of hardware and software.
- Determine the response time to online user requests.
- Determine transaction processing turnaround time.

How to Use Execution Testing

Execution testing can be conducted in any phase of the system development life cycle. The testing can evaluate a single aspect of the system, for example, a critical routine in the system, or the ability of the proposed structure to satisfy performance criteria. Execution testing can be performed in any of the following manners:

- Using hardware and software monitors
- Simulating the functioning of all or part of the system using a simulation model
- Creating a quick and dirty program(s) to evaluate the approximate performance of a completed system

Execution testing may be executed onsite or off-site for the performance of the test. For example, execution testing can be performed on hardware and software before being acquired, or may be done after the application system has been completed. The earlier the technique is used, the higher the assurance that the completed application will meet the performance criteria.

Execution Test Examples

Examples of the use of execution testing include:

- Calculating turnaround time on transactions processed through the application
- Determining that the hardware and software selected provide the optimum processing capability
- Using software monitors to determine that the program code is effectively used

When to Use Execution Testing

Execution testing should be used early in the developmental process. While there is value in knowing that the completed application does not meet performance criteria, if that assessment is not known until the system is operational, it may be too late or too costly to make the necessary modifications. Therefore, execution testing should be used at that point in time when the results can be used to affect or change the system structure.

Recovery Testing Technique

Recovery is the ability to restart operations after the integrity of the application has been lost. The process normally involves reverting to a point where the integrity of the system is known, and then reprocessing transactions up until the point of failure. The time required to recover operations is affected by the number of restart points, the volume of applications run on the computer center, the training and skill of the people conducting the recovery operation, and the tools available for recovery.

Objectives

Recovery testing is used to ensure that operations can be continued after a disaster. Recovery testing not only verifies the recovery process, but also the effectiveness of the component parts of that process. Specific objectives of recovery testing include:

- Preserve adequate backup data.
- Store backup data in a secure location.

- Document recovery procedures.
- Assign and train recovery personnel.
- Develop recovery tools and make available.

How to Use Recovery Testing

Recovery testing can be conducted in two modes. First, the procedures, methods, tools, and techniques can be assessed to evaluate whether they appear adequate; and second, after the system has been developed, a failure can be introduced into the system and the ability to recover tested.

Evaluating the procedures and documentation is a process using primarily judgment and checklists. On the other hand, the actual recovery test may involve off-site facilities and alternate processing locations. Testing the procedures is normally done by skilled systems analysts, professional testers, or management personnel. On the other hand, testing the actual recovery procedures should be performed by computer operators and other clerical personnel, who would be involved had there been an actual disaster instead of a test disaster.

A simulated disaster is usually performed on one aspect of the application system. For example, the test may be designed to determine whether people using the system can continue processing and recover computer operations after computer operations cease. While several aspects of recovery need to be tested, it is better to test one segment at a time rather than induce multiple failures at a single time. When multiple failures are induced, and problems are encountered, it may be more difficult to pinpoint the cause of the problem than when only a single failure is induced.

It is preferable not to advise system participants when a disaster test will be conducted. For example, a failure might be intentionally introduced during a normal system test to observe reaction and evaluate the recovery test procedures. When people are prepared, they may perform the recovery test in a manner different from the performance when it occurs at an unexpected time. Even if the participants know that recovery may be part of the test, it is not recommended to let them know specifically when it will occur, or what type of recovery will be necessary.

Recovery Test Example

Recovery testing can involve the manual functions of an application, loss of input capability, loss of communication lines, hardware or operating system failure, loss of database integrity, operator error, or application system failure. It is desirable to test all aspects of recovery processing. Some specific examples of recovery testing include:

- Inducing a failure into one of the application system programs during processing. This could be accomplished by inserting a special instruction to look for a transaction code that upon identification would cause an abnormal program termination.

- The recovery could be conducted from a known point of integrity to ensure that the available backup data was adequate for the recovery process. When the recovery had been completed, the files at the point where the exercise was requested could be compared to the files recreated during the recovery process.

When to Use Recovery Testing

Recovery testing should be performed whenever the user of the application states that the continuity of operation of the application is essential to the proper functioning of the user area. The user should estimate the potential loss associated with inability to recover operations over various time spans; for example, the inability to recover within five minutes, one hour, eight hours, and a week. The amount of the potential loss should both determine the amount of resource to be put into disaster planning as well as recovery testing.

Operations Testing Technique

After testing, the application will be integrated into the operating environment. At this point in time, the application will be executed using the normal operation staff, operations procedures, and documentation. Operations' testing is designed to verify prior to production that the operating procedures and staff can properly execute the application.

Objectives

Operations' testing is primarily designed to determine whether the system is executable during normal systems operations. The specific objectives include:

- Determine the completeness of computer operator documentation.
- Ensure that the necessary support mechanisms, such as job control language, are prepared and function properly.
- Evaluate the completeness of operator training.
- Test to ensure that operators using prepared documentation can, in fact, operate the system.

How to Use Operations Testing

Operations' testing evaluates both the process and the execution of the process. During the requirements phase, operational requirements can be evaluated to

determine the reasonableness and completeness of those requirements. During the design phase, the operating procedures should be designed and thus can be evaluated. This continual definition of the operating procedures should be subjected to continual testing.

The execution of operations testing can normally be performed in conjunction with other tests. However, if operations' testing is included, the operators should not be prompted or helped by outside parties during the test process. The test needs to be executed as if it was part of normal computer operations in order to adequately evaluate the effectiveness of computer operators in running the application in a true-to-life operations environment.

Operations Testing Example

Operations' testing is a specialized technical test of executing the application system and includes:

- Determining that the operator instructions have been prepared and documented in accordance with other operations instructions, and that computer operators have been trained in any unusual procedures
- Testing that the job control language statements and other operating systems support features perform the predetermined tasks
- Verifying that the file labeling and protection procedures function properly

When to Use Operations Testing

Operations' testing should occur prior to placing any application into a production status. If the application is to be tested in a production-type setting, operations testing can piggyback that process at a very minimal cost. It is as important to identify an operations flaw as it is an application flaw prior to placing the application into production.

Compliance Testing Technique

Compliance testing verifies that the application was developed in accordance with information technology standards, procedures, and guidelines. The methodologies are used to increase the probability of success, to enable the transfer of people in and out of the project with minimal cost, and to increase the maintainability of the application system. The type of testing conducted varies on the phase of the systems development life cycle. However, it may be more important to compliance test adherence to the process during requirements than at later stages in the life cycle because it is difficult to correct applications when requirements are not adequately documented.

Objectives

Compliance testing is performed to both ensure compliance to the methodology and to encourage and help the information technology professional comply with the methodology. Specific compliance objectives include:

- Determine that systems development and maintenance methodologies are followed.
- Ensure compliance to departmental standards, procedures, and guidelines.
- Evaluate the completeness and reasonableness of application system documentation.

How to Use Compliance Testing

Compliance testing requires that the prepared document or program is compared to the standards for that particular program or document. A colleague would be the most appropriate person to do this comparison. The most effective method of compliance testing is the inspection process.

Compliance Testing Examples

A peer group of programmers would be assembled to test line-by-line that a computer program is compliant with programming standards. At the end of the peer review, the programmer would be given a list of noncompliant information that would need to be corrected.

When to Use Compliance Testing

Compliance to information technology application system development standards and procedures is dependent upon management's desire to have the procedures followed and the standards enforced. Therefore, if management really wants compliance they should perform sufficient tests to determine both the degree of compliance with the methodology and to identify violators for management action. However, lack of compliance should also be used from the perspective that the standards may be misunderstood, not adequately instructed or publicized, or may, in fact, be poor standards inhibiting the development of application systems. In these instances, it may be desirable to change the methodology.

Security Testing Technique

Security is a protection system that is needed for both secure confidential information and for competitive purposes to assure third parties their data will be

protected. The amount of security provided will be dependent upon the risks associated with compromise or loss of information. Protecting the confidentiality of the information is designed to protect the resources of the organization. However, information such as customer lists or improper disclosure of customer information may result in a loss of customer business to competitors. Security testing is designed to evaluate the adequacy of the protective procedures and countermeasures.

Objectives

Security defects do not become as obvious as other types of defects. Therefore, the objectives of security testing are to identify defects that are very difficult to identify. Even failures in the security system operation may not be detected, resulting in a loss or compromise of information without the knowledge of that loss. The security testing objectives include:

- Determine that adequate attention is devoted to identifying security risks.
- Determine that a realistic definition and enforcement of access to the system is implemented.
- Determine that sufficient expertise exists to perform adequate security testing.
- Conduct reasonable tests to ensure that the implemented security measures function properly.

How to Use Security Testing Techniques

Security testing is a highly specialized part of the test process. Most organizations can evaluate the reasonableness of security procedures to prevent the average perpetrator from penetrating the application. However, the highly skilled perpetrator using sophisticated techniques may use methods undetectable by novices designing security measures and/or testing those measures.

The first step in testing is the identification of the security risks and the potential loss associated with those risks. If either the loss is low or the penetration method mere routine, the information technology personnel can conduct the necessary tests. On the other hand, if either the risks are very high or the technology that might be used is sophisticated, specialized help should be acquired in conducting the security tests.

Security Test Example

Security testing involves a wide spectrum of conditions. Testing can first be divided into physical and logical security. Physical deals with the penetration by people in order to physically gather information, while logical security deals with

the use of computer processing and/or communication capabilities to improperly access information. Second, access control can be divided by type of perpetrator, such as employee, consultant, cleaning or service personnel, as well as categories of employees. The type of test conducted will vary upon the condition being tested and can include:

- Determination that the resources being protected are identified, and access is defined for each resource. Program or individual can define access.
- Evaluation as to whether the designed security procedures have been properly implemented and function in accordance with the specifications.
- Unauthorized access can be attempted in online systems to ensure that the system can identify and prevent access by unauthorized sources.

When to Use Security Testing

Security testing should be used when the information and/or assets protected by the application system are of significant value to the organization. The testing should be performed both prior to the system going into an operational status and after the system is placed into an operational status. The extent of testing should depend on the security risks, and the individual assigned to conduct the test should be selected based on the estimated sophistication that might be used to penetrate security.

References

Guide – CSTE Common Body Of Knowledge, V6.1